

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE PATENT TRIAL AND APPEAL BOARD

Appeal No. <b>2011-004918</b> Application No. <b>09/553,337</b>	
Title: METHOD FOR UPDATING SOFTWARE	
Inventor: Richard R. Reisman	Confirmation No. 5134
Filing Date: April 20, 2000	Examiner: Tammara R. Peyton
Attorney Docket No. 47003/0005	Group Art Unit: 2184

**REQUEST FOR REHEARING**

TO THE COMMISSIONER FOR PATENTS:

Pursuant to 37 C.F.R. § 41.52, the Applicant hereby requests rehearing of the decision of the Patent Trial and Appeal Board (the “Board”) rendered November 30, 2012. This Request for Rehearing is timely filed, and no fee is due. If, however, the Office believes that any fee is due, the Office is authorized to charge deposit account no. 50-5836.

## TABLE OF CONTENTS

I.	Procedural History .....	1
II.	Summary of Argument .....	1
III.	Background .....	1
	A. Claimed Invention .....	1
	B. The Kleinerman Reference .....	3
	C. The Examiner's Position .....	4
IV.	Points Misapprehended or Overlooked by the Board .....	6
	A. The Board Decision Inaccurately Describes Kleinerman .....	6
	B. The Board Decision Miscomprehends the Examiner's Rejection .....	7
	C. The Board's Rationale for Attributing the "In the Background" Limitation to Kleinerman is Factually Incorrect .....	10
	1. The "Seamless Migration" Mentioned in Kleinerman in no Way Suggests that His API Operates in the Background .....	10
	2. The Fact That Kleinerman's Host Computer is Multitasking Does Not Suggest That the API Operates in the Background .....	11
	3. Kleinerman's "Invisible" Session Window, Which Refers to Session-Level Communication, is Unrelated to Whether the API Operates in the Background .....	12
V.	Conclusion .....	13

## **I. Procedural History**

In a final rejection dated January 25, 2010 (the “Final Rejection”), the Examiner rejected all pending claims under 35 U.S.C. § 103 as allegedly being obvious over a combination of at least four references, including U.S. Patent No. 5,734,871 (“Kleinerman”). The Applicant appealed, filing an appeal brief (the “Appeal Brief”) on August 18, 2010. The Examiner lodged an answer (the “Examiner’s Answer”) on November 12, 2010, and the Applicant filed a reply (the “Reply Brief”) on January 12, 2011. The Board issued a seven-page (including the cover page) decision (the “Board Decision”) nearly two years later on November 30, 2012, affirming all rejections.

## **II. Summary of Argument**

The key issue for rehearing is whether Kleinerman discloses or suggests claim language reciting “an application programming interface (API)” and “wherein . . . instructions receive via the API a response to the functional request from the online service in the background, thereby permitting the graphical user interface to continue operation” (quoting representative claim 114 with emphasis added). Unfortunately, the Board Decision concluded that Kleinerman discloses or suggests the “in the background” limitation for reasons different from those advanced by the Examiner and understood by the Applicant. In so doing, the Board has misapprehended key content of Kleinerman and reached a conclusion that is contrary to the factual record. The Applicant therefore respectfully requests that rehearing is appropriate to reverse and/or remand the Examiner’s rejections.

## **III. Background**

### **A. Claimed Invention**

Independent claim 114 is representative for purposes of this Request for Rehearing and reads as follows (with emphases and reference labels added):

114. A computer program product comprising a tangible computer-readable medium having instructions stored thereon, the instructions comprising:

[A] first instructions, executable at a user station, for selecting among a plurality of available online services to support an application function, wherein the first instructions form an application programming interface (API) configured to provide a generic client interface for communicating a functional request associated with the application function to any one of the plurality of available online services;

[B] second instructions, executable at the user station, for directing the establishment and use of a communication link between the user station and an online service selected from the plurality of available online services; and

[C] third instructions, executable at the user station, for presenting a graphical user interface, generating the functional request, and communicating the functional request to the online service using the API,

[D] wherein portions of the third instructions are downloaded from the online service, and

[E] wherein the third instructions receive via the API a response to the functional request from the online service *in the background*, thereby permitting the graphical user interface to continue operation.

As can be seen, claim 114, refers to “an application program interface (API).” APIs were known in the art at the time of the Applicant’s invention, and the Applicant does not claim an API *per se*. Instead, an API having particular functionality is an element of the claimed invention in combination with other elements.

Specifically, the API, as claimed as part of the overall invention, is associated with three functions expressed in the parts of the claim labeled A, C, and E above. First, as expressed in part A, the API is “configured to provide a generic client interface for communicating a functional request associated with the application function to any one of the plurality of available online services.” Second, the “third instructions,” according to part

C of the claim, “us[e] the API” to “present[] a graphical user interface, generat[e] the functional request, and communicat[e] the functional request to the online service.” Finally and most importantly for purposes of this Request for Rehearing, part E of the claim further specifies and limits the operation of the API in the claimed invention by reciting that “the third instructions receive via the API a response to the functional request from the online service in the background, thereby permitting the graphical user interface to continue operation” (emphasis added).

## **B. The Kleinerman Reference**

Kleinerman is from an era when there was a clear dichotomy between mainframe or central computers, on one hand, and personal computers and workstations, on the other hand. (*See generally* Kleinerman 1:21 – 5:17.<sup>1</sup>) Kleinerman observes that the most powerful application programs were then written for mainframe or central computers, which had the requisite “high degree of storage capacity, processing capacity, and information management services.” (*Id.* at 1:44-46.) However, Kleinerman complains that such programs had custom user interfaces, which required special training to use, due to their non-standard nature. (*Id.* at 4:48-58.) Kleinerman therefore sought to combine the user friendliness and standardization of user interfaces featured on personal computers with the greater processing capabilities of software running on a mainframe host. (*Id.* at 5:20-31.) To this end, Kleinerman teaches a technique for a mainframe or central “host” computer to interact with a “secondary” personal computer, whereby the host can perform a variety of data processing and storage software functions, while the secondary computer presents a standardized user interface to those software packages running on the host. (*Id.* at 5:20, Fig. 1.) To accomplish this, Kleinerman teaches middleware modules that he calls “WHOOOP” and “AIM,” which run on the host and the secondary computers, respectively, and communicate with each

---

<sup>1</sup> All references to patents herein are in the form X:Y-Z, which means lines Y through Z, inclusive, of column X.

other to translate between host user interface formats and secondary computer user interface formats. (*Id.* at 5:65 – 6:43.) The Appeal Brief aptly summarizes Kleinerman as follows:

Kleinerman is directed to a method and apparatus for controlling the execution of an application in a host computer that is under the control of a secondary computer. (Kleinerman, 5:20-64.) More specifically, Kleinerman purports to allow a user to work with an application in the host computer via a user interface at the secondary computer. (*Id.*) The interaction between the user interface at the secondary computer and the application residing on the host computer is carried out, at least in part, by an Application Interface Module (AIM), an application program interface, and a component referred to as the Watch Host Patterns (WHOOOP). (Kleinerman, 5:65-6:43, 7:51-54.) The AIM resides on the secondary computer and communicates with the WHOOOP, residing on the host computer, through the application program interface. (*Id.*) Kleinerman, at most, describes the functionality of the application program interface as “[a] mechanism to pass information from the WHOOOP to the AIM, to manage displays at the [secondary computer], and to accept input from the secondary computer.” (Kleinerman, 7:51-54.)

(Appeal Brief at 12.)

### **C. The Examiner’s Position**

In finally rejecting the claims, the Examiner contended that Kleinerman’s API corresponds to the claimed “API” and that Kleinerman’s API would provide “seamless interaction” and therefore satisfy the claim language regarding “communicating functional requests to any one of a plurality of available online services,” as expressed in the part of claim 114 labeled A above. (Final Rejection at 9; Examiner’s Answer at 10.) In doing so, the Examiner relied on official notice (Final Rejection at 9; Examiner’s Answer at 10), even though it is not clear exactly what fact the Examiner was noting, and the Examiner misstated

the law and Office policy regarding Official Notice.<sup>2</sup> Next, the Examiner contended that “this seamless interaction would not exclude having a generic interface,” seemingly to reach the Applicant’s claim language referring to “a generic client interface” in part A of claim 114. (Examiner’s Answer at 10; *see also* Final Rejection at 9-10.)

Regarding the “in the background” limitation, the Examiner interpreted that phrase to mean “completely invisible to or transparent to the user of a program running on their [sic] system.” (Final Rejection at 10; Examiner’s Answer at 11-12.) Next, the Examiner stated that she “could find no statement in Kleinerman to the effect that the API is not ‘running in the background’ . . . .” (Examiner’s Answer at 12; *see also* Final Rejection at 11.) Because she could find no express teaching to the contrary, the Examiner reasoned that it is possible that Kleinerman’s API “could” run in the background:

Therein [sic], Examiner is taking the position that some of the programs of Kleinerman’s [sic] *could* run “in the background” and the interpretation of the broad phrase “in the background” is sufficient to preclude patentability as claimed and defined by Appellant.

---

<sup>2</sup> The Examiner misstates the law and Office policy regarding official notice, asserting, “The Office now requires applicants to provide persuasive evidence and/or arguments directly refuting the Official Notice before a supporting evidence is to be supplied by the examiner” and citing to MPEP §2144.03(C). (Examiner’s Answer at 10.) This is a misreading of MPEP § 2144.03(C), which requires that an applicant merely make a specific and well-grounded “demand . . . for the examiner to produce authority” (quoting *In re Chevenard*, 139 F.2d 711, 713 (CCPA 1943)). At all times, the burden remains on the Office to establish a *prima facie* case of unpatentability, and “Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well-known.” MPEP § 2144.03(A) (emphases added). If the Examiner’s position were correct, examiners could liberally take official notice of all material claim limitations and effectively shift the burden to applicants to demonstrate patentability in all cases. The Applicant reserves all rights to challenge the Office’s misuse of Official Notice via judicial review, if necessary. (*See* Appeal Brief at 13-14.)

(Examiner's Answer at 12 (emphasis added); *see also* Final Rejection at 11.) In essence, the Examiner premised the rejection on the mere possibility that Kleinerman's API might operate "in the background" simply because Kleinerman does not say that his API does not operate "in the background."

The Examiner's position is contrary to well-established black-letter law. A rejection cannot be based on mere possibilities regarding a prior art reference. The claimed features must either be explicitly called out in the prior art references or, in limited circumstances, inherent within the references. To rely on inherent disclosure in a prior art reference, the Office must establish that "the prior art necessarily functions in accordance with, or includes, the claimed limitations." *MEHL/Biophile Int'l Corp. v. Milgraum*, 192 F.3d 1362, 1365 (Fed. Cir. 1999) (emphasis added). But if the feature "is not inevitably present" in "the natural result flowing from the operation [of the prior art] as taught," it is not inherent. *In re Oelrich*, 666 F.2d 578, 581-82 (C.C.P.A. 1981) (emphasis added). "Where support must be based on an inherent disclosure, it is not sufficient that a person following the disclosure might obtain the result set forth in the [claim]; it must invariably happen." *Gubelmann v. Gang*, 408 F.2d 758, 766 (C.C.P.A. 1969) (emphasis added). Speculation cannot fill gaps in factual support. "[T]he speculative notion that by happenstance the [prior art feature] might, under hypothetical circumstances, be capable of operating as [claimed] is an insufficient basis" for inherency. *Bettcher Indus., Inc. v. Bunzl USA, Inc.*, 661 F.3d 629, 640 (Fed. Cir. 2011). Simply put, "[i]nherency does not embrace probabilities or possibilities." *Trintec Indus., Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 1297 (Fed. Cir. 2002); *see also* MPEP § 2112 IV (same).

#### **IV. Points Misapprehended or Overlooked by the Board**

##### **A. The Board Decision Inaccurately Describes Kleinerman**

The Board Decision inaccurately describes Kleinerman in several respects, stating as follows (with emphases added):



Kleinerman relates to the execution of computer programs. (Col. 1, ll. 24-25.) Kleinerman explains that one or more computer application programs (i.e., the claimed “functional request”) are simultaneously executed in one or more host computer systems (i.e., the claimed “client”) under the control of a second computer system that performs operations on data and instructions (i.e., the claimed “the plurality of available online services”) (Abstract), which includes use of an Application Program Interface (API) for controlling devices (col. 18, ll. 29-31).

(Board Decision at 4.) There are several inaccuracies in the Board’s description. First, the Board confuses which computers in Kleinerman are the client and the host. Kleinerman’s “host computer system” (or simply “HOST” as shown on the right side of Figure 1) is not a “client,” as stated in the Board Decision. It is clearly a host, not a client. Second, there is no “claimed ‘client’” in claim 114, the only claim discussed in the Board Decision. Instead, claim 114 refers to “a generic client interface,” which is provided on a “user station.” Third, the Board wrongly attributes “the claimed ‘the plurality of available online services’” to Kleinerman’s “second computer system.” That is plainly contrary to what Kleinerman teaches. Kleinerman teaches that the “host computer system” or “HOST” – not the client-like “second computer system” – provides mainframe services or functionality. Indeed, that is how the Examiner formulated the rejection, and the Applicant has responded accordingly.

The Board’s errors in describing Kleinerman cannot be dismissed as mere sloppiness in the drafting of an opinion. Unfortunately, these errors are symptomatic of a failure to appreciate fully the teachings of Kleinerman, as explained further below.

#### **B. The Board Decision Miscomprehends the Examiner’s Rejection**

In affirming the rejection, the Board appears to misunderstand the Examiner’s reasoning and rationale supporting the rejection. The Board first agreed with the Examiner that Kleinerman’s API corresponds to the claimed API. (Board Decision at 3-4.) Next, the Board agreed with the Examiner that Kleinerman’s API is “configured to provide a generic

client interface for communicating a functional request associated with the application function to any one of the plurality of available online services,” as claimed, although it is not clear whether the Board found this limitation to be satisfied by Kleinerman alone (*see id.* at 3-4 (“We agree with the Examiner” that Kleinerman alone teaches this limitation)) or Kleinerman in combination with three other references (*see id.* at 4 (“[W]e agree with the Examiner that the combination of Kleinerman, RIPscript, Microsoft, and Zellweger teaches the limitation”)).

Next, the Board agreed with the Examiner’s interpretation of “background” as “completely invisible to or transport to the user” (*id.* at 5) and reasoned that Kleinerman’s middleware software did operate in the “background,” as so interpreted.<sup>3</sup> To support that conclusion, the Board cited three facts from Kleinerman: (1) Kleinerman provided “seamless migration,” (2) Kleinerman’s host system is multitasking, and (3) Kleinerman describes a session window that is “invisible to the user”:

Because Kleinerman describes a seamless migration for the user in the host system, a multitasking host computer, and the creation of a session window “invisible to the user” during the interaction between the AIM and the host system, Kleinerman teaches the limitation “a response to the functional request from the online service in the background, thereby permitting the graphical user interface to continue operation.”

(Board Decision at 5-6.)

---

<sup>3</sup> The Applicant in no way concedes to the Office’s interpretation of the phrase “in the background.” As stated in the Reply Brief, the Applicant contends that the phrase in this context means “allowing a user interface to continue operation in between the time a functional request is sent to an online service and a response to the functional request is received from the online service . . . .” (Reply Brief at 4-5.) The Applicant reserves all rights to advocate this interpretation and to introduce evidence that Kleinerman fails to meet that definition (*e.g.*, that the Hypercard™ example starting at column 16, line 35 of Kleinerman did not support background tasks) on remand or via judicial review, if necessary. Regardless, the Board Decision cannot stand even under the Examiner’s mistaken proposed interpretation of “in the background.”

This reasoning by the Board, while mistaken in its own right as explained below, is quite different from that provided by the Examiner to argue that the “in the background” limitation was met by Kleinerman. Recall that the Examiner contended that Kleinerman taught the “in the background” limitation simply because it was possible that Kleinerman’s API might run in the background and Kleinerman did not expressly teach against that possibility. *See* § III-C *supra* at 5-6. While the Examiner cited to Kleinerman’s references to “seamless migration” and the like, those citations were to support the Examiner’s contention that Kleinerman taught other limitations relating to the claimed API – the ones in parts A and C of claim 114, not the “in the background” limitation in part E of the claim. *See* § III-C *supra* at 4-5. The Board has confused different parts of the claim and different aspects of the rejection.

In fact, as compared to the Examiner’s rejection, the Board has changed which portions of Kleinerman the Office alleges to correspond to the claim limitations in such a way that is tantamount to a new ground of rejection. The proper result, assuming *arguendo* that the Board’s new rationale were not patently flawed, would be to remand the application to the Examiner to enter a new ground of rejection and thereby give the Applicant an opportunity to respond to the newly formulated rejection. *See* MPEP § 1207.03(III) (“There is no new ground of rejection when the basic thrust of the rejection remains the same such that an appellant has been given a fair opportunity to react to the rejection.” (emphasis added)). However, as explained in the following subsection, remand would be futile because the Board’s new interpretation of Kleinerman is clearly flawed.

Further evidence of the Board’s misunderstanding of the Examiner’s rejection comes from the Board’s treatment of the Official Notice issue. The Board did not reach the issue of whether the Examiner’s reliance on Official Notice was proper because the Board reasoned (incorrectly) that Kleinerman explicitly taught the “in the background” limitation. (*See* Board Decision at 6, n. 1.) However, the Examiner invoked Official Notice with respect to

the API limitations recited in part A of the claim – not part E, where the “in the background” limitation appears.<sup>4</sup>

**C. The Board’s Rationale for Attributing the “In the Background” Limitation to Kleinerman is Factually Incorrect**

As explained above, the Board’s rationale for concluding that Kleinerman’s API operates “in the background” was grounded in three alleged aspects of Kleinerman: (1) “seamless migration,” (2) multitasking in the host, and (3) an “invisible” session window. None of these alleged teachings individually supports the conclusion that Kleinerman’s API operates “in the background,” and collectively they fall far short of suggesting that Kleinerman’s API operates “in the background” at all, let alone “receiv[ing] via the API a response to the functional request from the online service in the background, thereby permitting the graphical user interface to continue operation,” as claimed.

**1. The “Seamless Migration” Mentioned in Kleinerman in no Way Suggests that His API Operates in the Background**

First, the “seamless migration” mentioned in Kleinerman in no way suggests that the API operates in the background. Instead, Kleinerman’s only reference to “seamless’ migration” refers to the ease with which users can adapt to different application programs running on the host or updates to a particular application program running on the host, because Kleinerman provides a standard type of user interfaces for all host programs:

---

<sup>4</sup> In fairness to the Board, the Examiner’s positions are so vaguely and confusingly stated in both the Final Rejection and the Examiner’s Answer that it is challenging and time consuming to carefully analyze the rejections to understand clearly the Examiner’s positions and supporting reasoning regarding each claim limitation. Still, it is the duty of the Board to review rejections, even one confusingly stated, and it is improper, as well as unfair, to penalize an applicant for confusion caused by an examiner.

The user interface can apply common conventions and standards to a wide variety of applications. New users are thus shielded from the vagaries and inconsistencies of the host system. This reduces the time required to train personnel permits “seamless” migration for the user community across changes to the host system, applications and the host system application programs on the connectivity requirements between the host and the secondary computer.

(Kleinerman 9:16-25.) This “seamless’ migration” bears no logical relation to whether the API operates in the foreground or background. Logically, Kleinerman’s technology could provide this “seamless’ migration” while the API operates entirely in the foreground. The Board Decision provides no explanation for reaching a contrary conclusion. In fact, it was clear error for the Board to point to this “seamless’ migration” as supporting the Board’s conclusion that Kleinerman’s API operates in the background.

## **2. The Fact That Kleinerman’s Host Computer is Multitasking Does Not Suggest That the API Operates in the Background**

Second, the fact that the host computer in Kleinerman is multitasking does not suggest that the API operates in the background. There is simply no relation between host multitasking and whether the API operates in the foreground or background. To be sure, the Board Decision offers no explanation for such a relationship, and the Examiner never alleged such a relationship. Quite possibly, the host could multitask, servicing multiple secondary computers, while each secondary computer operates entirely in the foreground, with no background processing whatsoever. In fact, multitasking at the host computer in no way implies anything about behavior of an API at Kleinerman’s secondary computer. Perhaps, if Kleinerman taught that the secondary computers are multitasking, this argument might have some traction, but Kleinerman is clear and unambiguous in describing only his host as a multitasking computer. All references to multitasking in Kleinerman are to the host computer. (See Kleinerman 1:40:46, 2:18-21, 2:34-37, 2:44-47, 5:20-31, 6:28-32.) Simply put,

the fact that Kleinerman's host computer is multitasking does not in any way logically suggest that an API at the secondary computer is multitasking, let alone, operates in the background.

**3. Kleinerman's "Invisible" Session Window, Which Refers to Session-Level Communication, is Unrelated to Whether the API Operates in the Background**

Third, the "invisible" session window in Kleinerman refers to session-level communication between the host and secondary computer. (*See id.* at 18:3-8.) Naturally, the session level is below the application level visible to the user and therefore it is not surprising that the session window is invisible to the user by default. In fact, it would be unusual to make low-level communication details visible to the user, since such details are typically immaterial to the user. The fact that there is an invisible session layer for communication is unrelated to whether the API operates in the foreground or the background. Again, the Board Decision offers no explanation for such a relationship, and the Examiner never alleged such a relationship. It is quite possible that the API operates in the foreground while the session-level communication details are kept invisible to the user.

In all, none of the reasons cited by the Board for its conclusion that Kleinerman's API operates in the background is sound. Collectively, they have no greater force. It simply does not logically follow from (1) a multitasking host computer, (2) an invisible session-level communication process, and (3) "seamless migration" for user experiences across different host programs that Kleinerman's secondary computer's API operates in the background. The Board's reasoning in this respect is *non sequitur*.

## V. Conclusion

For the foregoing reasons, the Applicant respectfully requests that the Board rehear this appeal, reconsider its decision, and reverse all rejections. Alternatively, the Board should at least remand the application to the Examiner so that the Applicant can have a fair opportunity to respond to the new grounds of rejection, as articulated in the Board Decision.

Respectfully submitted,

Date: 2013 January 30

By: / M.C. Phillips /

Matthew C. Phillips

Registration No. 43,403

Renaissance IP Law Group, LLP

9600 SW Oak St., Suite 560

Portland, Oregon 97223

Telephone: (503) 964-1129

Facsimile: (503) 517-9919

Attorney Docket No. 47003/0005